

PATENT

AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated in the following listing of all claims:

1. (Currently Amended) A multipass parser implementation product comprising:  
plural miniparsers each successively operable on a respective abstract syntax tree that  
corresponds to an input information encoding and that includes transformations of  
predecessor ones, if any, of the miniparsers,  
wherein respective ones of the miniparsers are limited to particular subsets of syntactic  
constructs to be parsed in the input information encoding.
2. (Original) The multipass parser implementation of claim 1,  
wherein, for at least some of the miniparsers, the respective abstract syntax tree is an  
output of a respective predecessor one of the miniparsers.
3. (Original) The multipass parser implementation of claim 1,  
wherein, for at least some of the miniparsers, respective input and output abstract syntax  
trees are separately encoded.
4. (Original) The multipass parser implementation of claim 1,  
wherein the syntactic constructs are those of a programming language; and  
wherein the information encoding includes code substantially in accordance therewith.
5. (Original) The multipass parser implementation of claim 1,  
wherein operation of each of the miniparsers is on a substantial entirety of its respective  
abstract syntax tree.
6. (Original) The multipass parser implementation of claim 1,  
wherein the syntactic constructs are those defined by a grammar; but  
wherein none of the miniparsers individually implements the grammar.
7. (Original) The multipass parser implementation of claim 1,

## PATENT

wherein the syntactic constructs are those defined by a grammar; and  
wherein the operation of successive ones of the miniparsers corresponds to the grammar.

8. (Original) The multipass parser implementation of claim 1, wherein the plural miniparsers include:

- a comment parser;
- a delimiter parser;
- a top-level statement parser;
- a compilation unit parser; and
- a name parser.

9. (Original) The multipass parser implementation of claim 1, wherein the plural miniparsers correspond to an hierarchy of functional transformations.

10. (Original) The multipass parser implementation of claim 1, embodied as a computer program product encoded in at least one computer readable medium.

11. (Original) The multipass parser implementation of claim 10, wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium.

12. (Original) A software engineering tool including the plural miniparsers of claim 1.

13. (Original) A method of implementing a parser for an input information encoding, the method comprising:

- defining a succession of miniparsers each operable on a respective parse state resulting from a predecessor one of the miniparsers,
- wherein each of the miniparsers recognizes only a particular subset of syntactic constructs to be parsed in the input information encoding.

## PATENT

14. (Original) The method of claim 13,  
wherein operation of each successive miniparser transforms an output of a predecessor  
miniparser.
15. (Original) The method of claim 13,  
wherein operation of each successive miniparser successively refines a parse tree that  
corresponds to the input information encoding.
16. (Original) The method of claim 13,  
wherein operation of each successive miniparser produces a distinct parse tree that  
represents a successive refinement corresponding to the input information  
encoding.
17. (Original) The method of claim 13, further comprising:  
executing the miniparsers in succession.
18. (Original) The method of claim 13,  
wherein the syntactic constructs are those of a programming language; and  
wherein the input information encoding includes code substantially in accordance  
therewith.
19. (Original) The method of claim 13,  
wherein a first executed one of the miniparsers implements a lexer.
20. (Original) The method of claim 13,  
wherein a first executed one of the miniparsers operates on a lexed encoding  
corresponding to the input information.
21. (Original) The method of claim 13, applied to a compilation unit of program code,  
wherein an earlier executed one of the miniparsers associates comment tokens of a  
compilation unit with respective non-comment tokens; and  
wherein a later executed one of the miniparsers matches grouping tokens.

## PATENT

22. (Original) The method of claim 21,  
wherein a still later executed one of the miniparsers segregates tokens into top-level  
statements.

23. (Original) A method of parsing an information encoding, the method comprising:  
performing plural successive transformations, each successive one of the transformations  
operating on an abstract syntax tree that is a result of a predecessor one of the  
transformations,  
each of the successive transformations handling only a subset of syntactic constructs to  
be recognized in the information encoding.

24. (Original) A computer program product encoded in at least one computer readable  
medium and comprising:  
functional encodings of at least two miniparsers, a first one of the miniparsers executable  
to transform a first parse tree into a second parse tree and the second one of the  
miniparsers executable to transform the second parse tree into a third parse tree;  
each of the at least two miniparsers recognizing only a subset of syntactic constructs to be  
parsed in an information encoding to which the first, second and third parse trees  
correspond.

25. (Original) The computer program product of claim 24,  
wherein the at least one computer readable medium is selected from the set of a disk, tape  
or other magnetic, optical, or electronic storage medium and a network, wireline,  
wireless or other communications medium.

26. (Original) An apparatus comprising:  
encoded information substantially in accordance with a grammar;  
multipass means for performing plural successive transformations on the encoded  
information, each successive transformation handling only a subset of syntactic  
constructs in accordance with the grammar.